Applications of Context-Free Grammers

Gur Saran Adhar

Grammars are used to describe programming languages. Most importantly there is a mechanical way of turning the description as a Context Free Grammar (CFG) into a parser, the component of the compiler that discovers the structure of the source program and represents that structure as a tree.
For example, The *Document Type Definition* (DTD) feature of XML (Extensible Markup Language) is essentially a context-free grammar that describes the allowable HTML tags and the ways in which these tags may be nested. For example, one could describe a sequence of characters that was intended to be interpreted as a phone number by `<PHONE> and </PHONE>` .
Example-1: Typical programming languages use parentheses and or brackets in a nested and balanced fashion. That is, we must be able to match some left parenthesis against a right parenthesis that appears immediately to its right, remove both of them and repeat. If we eventually eliminate all the parenthesis, then the string was balanced. Example of strings with balanced parenthesis are (()), ()(), (()()), while)(, and (() are not balanced.
A grammar with the following productions generates all and only the strings with balanced parenthesis:

$$B \to BB \mid (B) \mid \lambda$$

The first production, $B \to BB$, says that concatenation of two strings of balanced parenthesis is balanced. That is, we can match the parenthesis in two strings independently.
The second production, $B \to (B)$, says that if we place a pair of parenthesis around a balanced string, then the result is balanced.
The third production, $B \to \lambda$ is the basis, which says that an empty string is balanced.
Example-2: There are numerous aspects of typical programming language that behave like balanced parentheses. Beginning and ending of code blocks, such as **begin** and **end** in Pascal, or the curly braces { ... } of C, are examples. There is a related pattern thar appears occasionally, where "parentheses" can be balanced with the exception that there can be a unbalanced left parentheses. An example is the treatment of **if** and **else** in C. An if-clause can appear unbalanced by any else-clause, or it may be balanced by a matching else-clause. A grammer that generates the possible sequence of **if** and **else** (represented by i and e, respectively) is:

$$S \to SS \mid iS \mid iSe \mid \lambda$$

For instance, $ieie$, iie, and iei are possible sequences of **if** and **else**'s and each of these strings is generated by the above grammer. Some examples of illegal sequences not generated by the grammer are, ei, $ieeii$, iee.

Example-3: We give below CFG that describes some parts of the structure of HTML (Hypertext Markup Language).

$Char \rightarrow a \mid A \mid \ldots$

$Text \rightarrow \lambda \mid Char\ Text$

$Doc \rightarrow \lambda \mid Element\ Doc$

$Element \rightarrow Text \mid < EM > Doc < /EM > \mid < P > Doc \mid < OL > List < /OL >$

$List \rightarrow \lambda \mid ListItem\ List$

$ListItem \rightarrow < LI > Doc$

Example-4: Let G be a grammar with the set of variables:

$$\begin{aligned} V \;=\; & \{S, < Noun\ phrase >, < Verb\ phrase >, < Adjective\ phrase >, \\ & < Noun >, < Verb >, < Adjective >\} \end{aligned}$$

the alphabet set:
$\Sigma = \{big, stout, John, bought, white, car, Jim, cheese, ate, green\}$ with the rules:

(1) $S \rightarrow < Noun\ phrase >< Verb\ phrase >$
(2) $< Noun\ phrase > \rightarrow < Noun > \mid < Adjective\ phrase >< Noun > \mid \lambda$
(3) $< Verb\ phrase > \rightarrow < Verb >< Noun\ phrase >$
(4) $< Adjective\ phrase > \rightarrow < Adjective\ phrase >< Adjective > \mid \lambda$
(5) $< Noun > \rightarrow John \mid car \mid Jim \mid cheese$
(6) $< Verb > \rightarrow bought \mid ate$
(7) $< Adjective > \rightarrow big \mid stout \mid white \mid green$

Then the grammar generates, in particular, the following strings:

John bought car
Jim ate cheese
big Jim ate green cheese
John bought big car
big stout John bought big white car
Unfortunately, the grammer also generates sentences like:
big stout car bought big stout car
big cheese ate Jim
green Jim ate green big Jim